SQL Window Functions

Monti Kr. Raj

Introduction to Window Functions

- Window functions perform calculations across a set of rows related to the current row.
- Unlike aggregate functions, they do not collapse rows.
- Useful for running totals, rankings, moving averages, etc.

Basic Syntax

```
FunctionName (expression) OVER (
  PARTITION BY column
  ORDER BY column
  ROWS or RANGE clause
)
```

Example:

```
SUM(salary) OVER (PARTITION BY department_id ORDER BY hire_date)
```

Common Window Functions

- ROW_NUMBER() – assigns a unique sequential integer.
- RANK() – assigns rank with gaps for ties.
- DENSE_RANK() – assigns rank without gaps.
- NTILE(n) – divides rows into n buckets.
- SUM(), AVG(), MIN(), MAX() – can be used as window functions.

• ROW_NUMBER( )

```
Select employee_id, department_id, salary,
row_number( ) Over (partition by department_id Order by salary desc) as row_num
from employees;
```

RANK() vs DENSE_RANK()

- RANK(): Skips ranks if there are ties.

  Example: 1, 2, 2, 4

- DENSE_RANK(): No gaps in ranks.

  Example: 1, 2, 2, 3

Running Totals

```
Select order_id, customer_id, amount,
sum(amount) Over (partition by customer_id order by order_date) as running_total
from orders;
```

Moving Average

```
Select order_date, amount,
avg(amount) Over (order by order_date rows between 2 preceding and current row) as moving_avg
from orders;
```

Conclusion

- Window functions enhance SQL analytics.
- Provide flexibility with PARTITION BY and ORDER BY.
- Useful for reporting, trend analysis, and data ranking.